ISSN (ONLINE) : 2395-695X
ISSN (PRINT) : 2395-695X
Available online at www.ijarbest.com

International Journal of Advanced Research in Biology, Ecology, Science and Technology (IJARBEST)
Vol. 1, Issue 9, December 2015

REDUCING DEFICIT QUEUE DELAY USING VIRTUAL MEMORY

N.Deepa[1], V.R.Arulmozhi[2], B.Vijaya Nirmala[3]
[1,2,3] *Assistant Professor/Department of CSE, RVS Educational Trust's Group of Institution Dindigul, Tamilnadu, India.*
[1] [1]deepanatrayan@gmail.com
[2] [2]arulmozhiram@gmail.com
*[3]*bvijayanirmalacse@gmail.com

Abstract

Scheduling is the method by which threads, processes or data flows are given access to system resources. This is usually done to load balance and share system resources effectively or achieve a target quality of service. The need for a scheduling algorithm arises from the requirement for most modern systems to perform multitasking and multiplexing. Request queues indicate the number of elastic requests, and deficit queues indicate the deficit in inelastic service. Caches are of finite size and can be refreshed periodically from a media vault. It considers two cost models that correspond to inelastic requests for streaming stored content and real time streaming of events respectively. It increases the server stability on the time overload of data by using the input size limitation concepts. It evaluates the problems about the data maintenance and data scheduling management.

Keywords: quality of service, multitasking, multiplexing, request queues

1. INTRODUCTION

The past few years have seen the rise of smart handheld wireless devices as ameans of content consumption. Content might include streaming applications in which chunks of the file must be received under hard delay constraints, as well as file downloads such as software updates that do not have such hard constraints. The core of the Internet is well provisioned, and network capacity constraints for content delivery are at the media vault (where content originates) and at the wireless access links at end-users. Hence, a natural location to place caches for a content distribution network (CDN) would be at the wireless gateway, which could be a cellular base station through which users obtain network access. Furthermore, it is natural to try to take advantage of the inherent broadcast nature of the wireless medium to satisfy multiple users simultaneously.

There are multiple cellular *base stations* (BSs), each of which has a cache in which to store content. The content of the caches can be periodically refreshed through accessing a *media vault*. We divide users into different *clusters*, with the idea that all users in each cluster are geographically close such that they have statistically similar channel conditions and are able to access the same base stations. Note that multiple clusters could be present in the same cell based on the dissimilarity of their channel conditions to different base stations. The requests made by each cluster are aggregated at a logical entity that we call a *front end* (FE) associated with that cluster. The front end could be running on any of the devices in the cluster or at a base station, and its purpose is to keep track of the requests associated with the users of that cluster. The following constraints affect system operation: 1) the wireless network between the caches to the users has finite capacity; 2) each cache can only host a finite amount of content; and 3) refreshing content in the caches from the media vault incurs a cost.

Users can make two kinds of requests, namely: 1) elastic requests that have no delay constraints, and 2) inelastic requests that have a hard delay constraint. Elastic requests are stored in a *request queue* at each front end, with each type of request occupying a particular queue. Here, the objective is to stabilize the queue, so as to have finite delays. For inelastic requests, we adopt the model proposed in [2] wherein users request chunks of content that have a strict deadline, and the request is dropped if the deadline cannot be met. The idea here is to meet a certain target *delivery ratio*, which could be something like "90% of all requests must be met to ensure smooth playout." Each time an inelastic request is dropped, a *deficit queue* is updated by an amount proportional to the delivery ratio. We would like the average value of the deficit to be zero.

We are interested in solving the joint content placement and scheduling problem for both elastic and inelastic traffic in wireless networks. In doing so, we will also determine the value of predicting the demand for different types of content and what impact it has on the design of caching algorithms.

Resource allocation is used to assign the available resources in an economic way. It is part of resource management. In project management, resource allocation is the scheduling of activities and the resources required by those activities while taking into consideration both the resource availability and the project time. Resource allocation may be decided by using computer programs applied to a specific domain to automatically and dynamically distribute resources to applicants. The benefits of the proposed system is that the virtual memory reduces deficit queue delay. It stabilizes the system load within the capacity region. It minimizes the average expected cost while stabilizing the deficit queues

2. RELATED WORK

The problem of caching, and content scheduling has earlier been studied for onlineWeb caching and distributed storage systems. A commonly used metric is a competitive ratio of misses, assuming an adversarial model. Examples of work in this context are [3]–[5]. Load balancing and placement with linear communication costs is examined in [6] and [7]. Here, the objective is to use distributed and centralized integer programming approaches to minimize the costs. However, this work does not take account for network capacity constraints, delay-sensitive traffic, or wireless aspects. The techniques that we will employ are based on the literature on scheduling schemes. Tassiulas *et al.* proposed the MaxWeight scheduling algorithm for switches and wireless networks in their seminal work [8]. They proved that this policy is throughput-optimal and characterized the capacity region of the single-hop networks as the convex hull of all feasible schedules. Various extensions of this work that followed since are [9]–[12].

These papers explore the delays in the system for single downlink with variable connectivity, multi rate links, and multi hop wireless flows. However, these do not consider content distribution with its attendant question of content placement. Closest to our work is [13], which, however, only considers elastic traffic and has no results on the value of prediction.

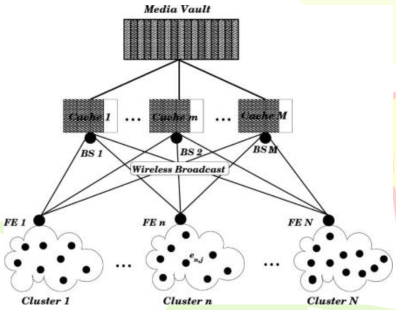

Fig. 1. Wireless content distribution.

An abstraction of such a network is illustrated in Fig. 1. There are multiple cellular *base stations* (BSs), each of which has a cache in which to store content. The content of the caches can be periodically refreshed through accessing a *media vault*. We divide users into different *clusters*, with the idea that all users in each cluster are geographically close such that they have statistically similar channel conditions and are able to access the same base stations. Note that multiple clusters could be present in the same cell based on the dissimilarity of their channel conditions to different base stations. The requests made by each cluster are aggregated at a logical entity that we call a *front end* (FE) associated with that cluster. The front end could be running on any of the devices in the cluster or at a base station, and its purpose is to keep track of the requests associated with the users of that cluster. The following constraints affect system operation:

1) the wireless network between the caches to the users has finite capacity;
2) each cache can only host a finite amount of content;
3) refreshing content in the caches from the media vault incurs a cost.

Users can make two kinds of requests..

The problem of caching, and content scheduling has earlier been studied for online Web caching and distributed storage systems.

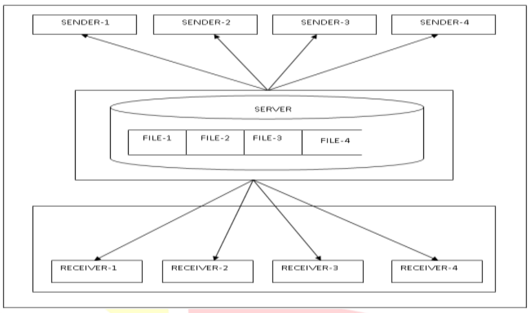

Fig 2. System Architecture

Network Data Transaction describes the data transaction from the sender to receiver through the server. The transferred data from the sender nodes, in different sizes, i.e., large and small capacity files. The server manages and stores the all types of data in storage. The sender chooses the specific location of the file to be sent to the receiver. It also mentions the Internet Protocol address of the receiver and also the receiving date and time.

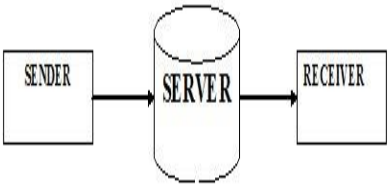

Fig 3. Network Data Transaction

Server Storage Process contains the various types of files in different sizes from the sender nodes. If all files are in same size means the server can balance the data easily. If the files in various sizes means, the server cannot balancing the loads of data for the transaction process. So that time, server needs a support memory for load balancing of data. It loads the files of the sender that are to be sent to the receiver.Data Scheduling Operations contains the server receives the data from the various types of nodes. The received files are stored in general storage When the distribution process starts, the server checks the time of the system with the receive time which is mentioned by the sender. If it matches, then the server sends the files to the receivers based on the time priority. When the schedule task is completed, the time server removes the schedule backup for server efficiency.

Data Distributions To Clusters contains the server stores the data in dispatching queue based on the file size. When the distribution process starts, the server sends the files to the grouping nodes conditions based on the time priority basis. The receiver sends the acknowledgement to the sender after it receives the data from the sender.

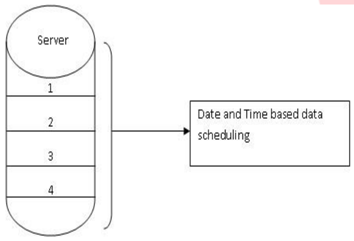

Fig 4. Data Scheduling Operations

The techniques that we will employ are based on the literature on scheduling schemes. Tassiulas *et al.* proposed the MaxWeight scheduling algorithm for switches and wireless networks in their seminal work [8].

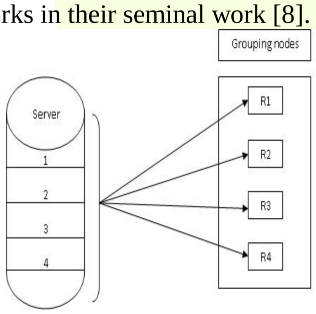

Fig 5. Data Distributions to clusters

They proved that this policy is throughput-optimal and characterized the capacity region of the single-hop networks as the convex hull of all feasible schedules.

Various extensions of this work that followed since are [9]–[12].These papers explore the delays in the system for single downlink with variable connectivity, multirate links, and multihop wireless flows. However, these do not consider content distribution with its attendant question of content placement. Closest to our work is [13], which, however, only considers elastic traffic and has no results on the value of prediction.

3. IMPLEMENTATION

In this paper, we develop algorithms for content distribution with elastic and inelastic requests. We use a request queue to implicitly determine the popularity of elastic content. Similarly, the deficit queue determines the necessary service for inelastic requests. Content may be refreshed periodically at caches. We study two different kinds of cost models, each of which is ap-propriate for a different content distribution scenario. The first is the case of file distribution (elastic) along with streaming of stored content (inelastic), where we model cost in terms of the frequency with which caches are refreshed. The second is the case of streaming of content that is generated in real-time, where content expires after a certain time, and the cost of placement of each packet in the cache is considered.

We first characterize the capacity region of the system and develop feasibility constraints that any stabilizing algo-rithm must satisfy. Here, by stability we mean that elastic request queues have a finite mean, while inelastic deficit values are zero on average.

We develop a version of the max-weight scheduling algo-rithm that we propose to use for joint content placement and scheduling. We show that it satisfies the feasibility con-straints and, using a Lyapunov argument, also show that it stabilizes the system of the load within the capacity region. As a by-product, we show that the value of knowing the ar-rival rates is limited in the case of elastic requests, while it is not at all useful in the inelastic case.

We next study another version of our content distribution problem with only inelastic traffic, in which each content has an expiration time. We assume that there is a cost for replacing each expired content chunk with a fresh one. For this model, we first find the feasibility region and, fol-lowing a similar technique to [14], develop a joint content placement and scheduling algorithm that minimizes the av-erage expected cost while stabilizing the deficit queues.

We illustrate our main insights using simulations on a simple wireless topology and show that our algorithm is indeed capable of stabilizing the system. We also propose two simple algorithms, which are easily implementable, and compare their performance to the throughput-optimal scheme. Consider the content distribution network depicted in Fig. 1. There is a set of base stations $\mathcal{M}$ and each base station is as-sociated with a cache. The caches are all connected to a media vault that contains all the content. The users in the system are di-vided into clusters based on their geographical positions, and we let $\mathcal{N}$ denote the set of these clusters.

Also, as discussed in the Introduction, there are front ends in each cluster, also denoted by whose purpose is to aggregate requests from the users. Time is slotted, and we divide time into *frames* consisting of D time-slots. Requests are made at the beginning of each frame. There are two types of users in this system—inelastic and elastic—based on the type of requests that they make. Re-quests made by inelastic users must be satisfied within the frame in which they were made. Elastic users do not have such a fixed deadline, and these users arrive, make a request, are served, and depart.

The base stations employ multiple access schemes (e.g., OFDMA), and hence each base station can support multiple simultaneous unicast transmissions, as well as a single broad-cast transmission. It is also possible to study other scenarios (e.g., multicast transmissions to subsets of users) using our framework. We adopt a slow-fading packet erasure model for the wireless channels. Accordingly, the channel between cache and user is modeled as a stochastic ON-OFF process which is over frames and the state does not change during frame known to scheduler.

We assume that pieces of content have the same size,and we call the unit of storage and transmission as chunk when a channnela is ON,it can be used to mtransmit at most one chunk(per slot).

Pure Unicast Elastic Scenario In this section, we assume there are only requests for elastic content. As noted in Section II, these requests are to be served using unicast communications. For notational convenience, we assume that transmissions are between base stations and front ends, rather than to the actual users making the requests determine the *capacity region*, which is the set of all feasible requests.

Note that this model, in which front ends have independent and distinct channels to the caches, differs from the previously studied wired caching systems (see, e.g., [13]) because the wireless channels are not always ON. Therefore, the placement and scheduling must be properly coordinated according to the channel states.

Throughput-Optimal Scheme

Since it is hard to realize an offline prediction, placement and scheduling scheme, we now study our system of elastic requests in a queueing context. The development here is similar to the traditional switch scheduling problem, as relevant to our model.. JOINT ELASTIC-INELASTIC SCENARIO

In this section, we study the general case where elastic and inelastic requests coexist in the system. Recall that the elastic requests are assumed to be served through *unicast* communications between the caches and front ends, while the base stations *broadcast* the inelastic contents to the inelastic users. We further assumed servers can employ OFDMA method to simultaneously transmit over their single broadcast and multiple unicast channels.

Although these two types of traffic do not share the access medium, all the content must share the common space in the caches. Consequently, we require an algorithm that jointly solves the elastic and inelastic scheduling problems. In this section,we first determine the general *capacity region* of the system and then present our algorithm.. Joint Elastic-Inelastic Capacity Region Similar to the case of elastic content, we use to denote the presence of inelastic content in cache during frame . Since the channel states do not change during a frame, and there is at most one request for inelastic content by each user , each cache may schedule to broadcast content at most once per frame. Joint Elastic-Inelastic Scenario

In This Section, We Study the general case where elastic and inelastic requests coexist in the system. Recall that the elastic requests are assumed to be served through *unicast* communications between the caches and front ends, while the base stations *broadcast* the inelastic contents to the inelastic users. We further assumed servers can employ OFDMA method to simultaneously transmit over their single broadcast and multiple unicast channels. Although these two types of traffic do not share the access medium, all the content must share the common space in the caches. Consequently, we require an algorithm that jointly solves the elastic and inelastic scheduling problems. In this section, we first determine the general *capacity region* of the system and then present our algorithm.

4. CONCLUSION

In this paper, By using resource allocation algorithm, it easily reduces the delay of deficit queues and stabilizes the deficit queues and achieves an average cost that is arbitrarily close to the minimum cost. It increases the server stability on the time overload of data by using the input size limitation concepts and evaluates the problems about the data maintenance and data scheduling management then it is to investigate and to implement load balancing of the server which has the ability to implement to a swap memory to reduce the load balancing problem which increases the server stability and will reduce the deficit queue delay.

REFERENCES

[1] N. Abedini and S. Shakkottai, “Content caching and scheduling in wireless broadcast networks with elastic and inelastic traffic,” in *Proc.IEEE WiOpt*, 2011, pp. 125–132.

[2] I. Hou, V. Borkar, and P. Kumar, “A theory of QoS for wireless,”in *Proc. IEEE INFOCOM*, Rio de Janeiro, Brazil, Apr. 2009, pp.486–494.

[3] R. M. P. Raghavan, *Randomized Algorithms*. NewYork,NY,USA:Cambridge Univ. Press, 1995.

[4] P. Cao and S. Irani, “Cost-awareWWWproxy caching algorithms,” in*Proc. USENIX Symp. Internet Technol. Syst.*, Berkeley, CA, Dec. 1997,p. 18.

[5] K. Psounis and B. Prabhakar, “Efficient randomized Web-cache replacement schemes using samples from past eviction times,”*IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 441–455, Aug. 2002.

[6] N. Laoutaris, O.T. Orestis, V.Zissimopoulos, and I. Stavrakakis, “Distributed selfish replication,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 17, no. 12, pp. 1401–1413, Dec. 2006.

[7] S. Borst, V. Gupta, and A. Walid, “Distributed caching algorithms for content distribution networks,” in *Proc. IEEE INFOCOM*, San Diego,CA, USA, Mar. 2010, pp. 1–9.

[8] L. Tassiulas and A. Ephremides, “Stability properties of constrained queueing systems and scheduling policies for maximum throughput inmultihop radio networks,” *IEEE Trans. Autom. Control*, vol. 37, no.12, pp. 1936–1948, Dec. 1992.

[9] X. Lin and N. Shroff, “Joint rate control and scheduling in multihop wireless networks,” in *Proc. 43rd IEEE CDC*, Paradise Islands, Bahamas, Dec. 2004, vol. 2, pp. 1484–1489.

[10] A. Stolyar, “Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm,” *Queueing Syst. Theory Appl.*, vol. 50,no. 4, pp. 401–457, 2005.

[11] A. Eryilmaz and R. Srikant, “Joint congestion control, routing, and MAC for stability and fairness in wireless networks,” *IEEE J. Sel.*
Areas Commun., vol. 24, no. 8, pp. 1514–1524, Aug. 2006.

[12] J. Jaramillo and R. Srikant, “Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic,” in *Proc.IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 1–9.

[13] M. M. Amble, P. Parag, S. Shakkottai, and L. Ying, “Content-aware caching and traffic management in content distribution networks,” in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 2858–2866.

[14] M. Neely, “Energy optimal control for time-varying wireless networks,” *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul.2006.

[15] F. Foster, “On the stochastic matrices associated with certain queueing processes,” *Ann. Math. Statist.*, vol. 24, pp. 355–360, 1953.

[16] M. Neely, “Energy optimal control for time varying wireless networks,”*IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul.2006.